Real-Time RGB-D Volumetric Fusion on Mobile Devices

Jinjiang You (jinjiany)
Yixin Fei (yixinf)
Peipei Zhong (peipeiz)

March 2, 2024

1 Topic and Motivation

Accurate and real-time camera localization and scene reconstruction are crucial for mixed reality, robotics, and autonomous vehicles. RGB-D SLAM techniques stand out, especially when compared to more sensor-intensive SLAM approaches, because they leverage cost-effective and compact sensors. The interest in RGB-D SLAM systems surged following the release of Microsoft's Kinect in 2010. RGB-D sensors, which combine a standard RGB camera with a depth sensor, enable the direct acquisition of depth information with reasonable accuracy in real time using affordable hardware.

In this project, we aim to implement a real-time RGB-D volumetric fusion system based on KinectFusion[3] on PC and deploy it to mobile devices like iPads, harnessing the capabilities of RGB-D cameras and IMUs which are already integrated into these devices. Compared with other SLAM algorithms, KinectFusion requires less computation resources but can still achieve robust reconstruction results in real-time, which is suitable for mobile devices. Besides, KinectFusion uses a volumetric representation for the reconstructed model whose size only depends on the scanned scene instead of scanning time, which is suitable for mobile devices. Our objective is to enhance the accessibility of 3D mapping technology, making it lightweight and user-friendly on mobile devices, thereby facilitating a more engaging and immersive user experience on mobile platforms.

2 Anticipated Work Scope

We expect to implement the KinectFusion algorithm on PC, port it to IOS devices, and make it real-time. We will implement a complete fusion pipeline including reading data from cameras, core fusion algorithms, and visualizing the reconstructed scene on the UI. Besides, we will also use the inertial data provided by IOS API to improve the accuracy of KinectFusion.

Our implementation will rely on OpenCV and ORB-SLAM3 [1] library to do some bottom-level algorithms like extracting image key points and descriptors. We will also use Eigen (for C++) or Numpy (for Python) to perform vector-math computing. Other algorithms like ICP, ray-casting, and optimization will be implemented on our own. Our implementation will also depend on NVIDIA CUDA (on PC) or Metal/Vulkan (on IOS) to perform parallel computing. Also, it is important to note that Apple does not allow users to directly access its LiDAR camera's API to read the raw data. Therefore, we will read the captured RGB-D data via ARKit [2].

3 Evaluation Methods

Since we decide to implement the fusion algorithm on both PC and IOS devices, there are two ways to evaluate our implementation. On PC, we will test on some famous SLAM datasets like the TUM RGB-D [4] dataset. We intend to use the automatic evaluation tool in the TUM RGB-D to evaluate the quality of the resulting maps. Additionally, we will analyze the algorithm's frame rate to demonstrate its capability for real-time operation. On IOS devices, we will use their depth cameras to capture real-world data and perform a fusion algorithm on a background thread to show that it can run in real-time on Mobile devices.

References

[1] C. Campos, R. Elvira, J. J. G´omez, J. M. M. Montiel, and J. D. Tard'os. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[2] A. Inc. Arkit 6 - augmented reality. URL `https://developer.apple.com/augmented-reality/arkit/`.

[3] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011. doi: 10.1109/ISMAR.2011.6092378.

[4] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.